Model Checking a Networked System Without the Network

Rachid Guerraoui

Maysam Yabandeh
Bugs

• Testing
  – Looking for the bugs

• Debugging
  – Fixing the found bugs
Testing

1. State
2. Transition

Local Model Checking
Model Checking (MC)

- While (...)
  - An old state $\rightarrow$ A new state
- Exhaustive exploration of State Space
[Classic] MC Distributed Systems

- While (...)  
  - An old state $\rightarrow$ A new state
[Classic] MC Distributed Systems

- While (...)  
  - An old state $\rightarrow$ A new state
[Classic] MC Distributed Systems

• While (...)
  – An old state $\Rightarrow$ A new state

• (Global) State:
  o System state, i.e., Local node states
  o + Network state

• Transition:
  o Message handler
  o Local events: timers
State Space Coverage

- Exponential Explosion
- Exhaustive
Online Model Checking [NSDI’09]
How about Paxos?

- Simple state space: one local event (propose)
Agenda

2. What is the problem?
3. Local Model Checking (LMC)
Global Model Checking

- Huge global state size
Local Global Model Checking (LMC)
Local Model Checking (LMC)

- Transition?
  - Local events: timer
  - Network events:

Network
LMC: Shared Network

Node d states

Valid?  Invalid?
LMC: Architecture

Distributed System Implementation

- Initial States
- Local States
- Handler Execution
- Network I

msg

Local Model Checking
LMC: Architecture

Distributed System Implementation

Initial States → Local States → Handler Execution → Network $I^+$ → Errors

Local States

Invariant Checking

Invariant

Invariant Checking → [false] → Errors

Local States

Messages

msg → msg → msg
LMC: Architecture

Distributed System Implementation

Initial States → Local States

Local States → Handler Execution

Handler Execution → Network I^+

Network I^+ → Errors

System State Creation

Invariant Checking

Invariant Checking → [false]

Local States → LS

Handler Execution → msg

Network I^+ → msg

Errors → Invariants
LMC: System State Creation

Global Model Checking

- GS0
- GS1
- GS2
- GS3

Local Model Checking

- SS0
- SS1
- SS2
- SS3

Local States
LMC: System State Creation

• System States =
  – Combination of all local states: LMC-GEN
  – Only when there is a chance of bug
    • Paxos: different chosen values: LMC-OPT

• Complete

• Invalid states → Unsound bug report
LMC: Architecture

Distributed System Implementation

Initial States

Local States

Handler Execution

System State Creation

Invariant Checking

Network I⁺

Errors

Invariants

Local States

msg

msg

[false]
LMC: Architecture

Distributed System Implementation

Initial States

Local States

System State Creation

Handler Execution

Network I+ & Soundness Verification

Invariant Checking

Errors

Local States

Invariant

[false]
Soundness Verification

Predecessor

Local Model Checking

23
Soundness Verification

Node 0

Si ➔ Sj ➔ Sk

Node 1

Sj ➔ Sl ➔ Sm

Node 2

Sn ➔ So ➔ Sp

System State

Partial Order

Total Order

Local Model Checking
Soundness Verification

• Input: set of event stacks
• Output: valid or invalid
• Greedy Algorithm:
  – While exist enabled event e = stack\(_i\).top()
    • Run e
    • stack\(_i\).pop()
  – Return valid if there is no event left
Evaluation

• Prototype using MaceMC [NSDI ‘07]
  – Mace: structured C++ programs
  – (De)Serialization of states
  – Handler boundary → events

• Paxos: Nodes agree on the same values
  – A very complex distributed algorithm
  – Lots of communication at each round

• State space: 3 Nodes, 1 Proposes

• B-DFS: memory efficient MC [NSDI ‘07]
  – Bounded Depth First Search
  – Maintain a DB of state hashes to avoid loops

• 3 GHz Intel® Pentium® 4 CPU, 1 MB L2 cache
Evaluation: Elapsed Time
Evaluation: # of states

![Graph showing the comparison of B-DFS, LMC-GEN-system, LMC-OPT-system, and LMC-local with respect to the total number of states and depth.](image)
Testing

• Paxos:
  – Injected a previously reported bug [Liu et al., NSDI’07]
  – RedisCOVERS THE Bug

• 1Paxos: an efficient variant that uses one acceptor
  – Found a new bug in initialization of acceptOR
Related Work

LMC = CA + MA

• **CA: Cartesian Abstraction** [Ball, et. al, 2001] [Flanagan and Qadeer, 2003][Henzinger et. al, 2003] [Malkis et. al, 2006] [Malkis et. al, 2010]

• Thread-modular MC
  – Local asserts $\Rightarrow$ no system state creation
  – Complete, Unsound $\Rightarrow$ Theorem proving vs. Testing

• **MA: Monotonic Abstraction** [Mitchell, 2002]
  – Shared Network
Summary

Local Global Model Checking

• Local states, Shared network state
• Postpones State Explosion Problem
• Optimistic
• Decouples Exploration, System State Creation, and Soundness Verification
  – In Parallel
  – LMC-OPT (generalize?)
Thank you ...